IFW

PATENT APPLICATION
Attorney Docket No. PD-203009
Customer No. 20991

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

M. EROZ et al.

Application No.: 10/613,824

Filed: July 3, 2003

Title: METHOD AND SYSTEM FOR ROUTING IN LOW DENSITY PARITY CHECK (LDPC) DECODERS

Group Art Unit: 2184

Examiner: Not Yet Assigned

Mail Stop Petition
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Attention: Special Program Examiner GAU

REQUEST FOR RECONSIDERATION OF PETITION TO MAKE SPECIAL PURSUANT TO 37 C.F.R. § 1.102

Dear Sir:

In response to the decision on petition mailed on June 2, 2004, Applicants respectfully request reconsideration of this Petition to advance examination of this application pursuant to the provisions of 37 C.F.R. § 1.102(d) and MPEP 708.02 (VIII).

VIII. (A) The Commissioner is hereby authorized to charge Deposit Account 50-0383 $110 for a one month extension fee in accordance with 37 C.F.R. § 1.17(a)(1). The $130 petition fee has already been paid in conjunction with the original filing of this Petition on February 27, 2004. Should the Commissioner determine that an additional fee is due, he is hereby authorized to charge the additional fee to Deposit Account 50-0383.

VIII. (B) All of the claims presented in the above-identified patent application are believed to be directed to a single invention. If the Examiner believes that the pending claims are directed to more than one invention, Applicants hereby agree to elect claims directed to a single invention, without traverse.

The present invention is directed to a method and system for routing in decoders for low density parity check (LDPC) codes. LDPC codes are a class of block error control codes that allow a communication system to approach the Shannon limit, which is the theoretical upper limit for data rate at a given signal to noise ratio. However, LDPC codes have not been widely deployed commercially because of their complexity and because very large blocks are required for effective use, thus causing storage problems. Therefore, it is an objective of the present invention to provide an approach for decoding structured LDPC codes to be used in a communication system that efficiently uses the LDPC codes to support high data rates without introducing greater complexity and that minimizes storage requirements.

The present invention addresses this objective by providing a combination of method and structure for decoding LDPC-coded signals that retrieves edge values associated with a structured parity check matrix used to generate the LDPC-coded signal, wherein the edge values specify relationships between bit nodes and check nodes, and wherein the edge values are stored according to a predetermined scheme that permits concurrent retrieval of a set of edge values. The predetermined scheme preferably specifies contiguous physical memory locations for the set of edge values.

In another aspect of the present invention, edge values having bit nodes of *n* degrees are stored in a first portion of the memory, and edge values having bit nodes of greater than *n* degrees are stored in a second portion of the memory. In yet another aspect of the present invention, the set of edge values is retrieved in a single clock cycle of a processor coupled to the memory, and is adjacent to a group of M bit nodes or M check nodes, where M is the number of parallel processing engines.

VIII. (C) A pre-examination search was conducted by a professional search firm. The search was conducted in Class 375, subclasses 261, 265, 268, 269, 272, 279, and 286; Class 714, subclasses 752, 758, 777, and 781; and on computer using Delphion, EPO ESPACE, and PTO databases; and on the Internet. In addition, an International Search Report for corresponding International Patent Application No. PCT/US03/21071 has been issued by the European Patent Office acting as the International Search Authority.

VIII. (D) The pre-examination search revealed the following references:
(i) U.S. Patent Application Publication No. US 2003/0104788 A1
(ii) U.S. Patent Application Publication No. US 2003/0074626 A1
(iii) U.S. Patent Application Publication No. US 2003/0023917 A1
(iv) Mansour et al., "Low-Power VLSI Decoder Architectures for LDPC Codes", Proceedings, IEEE International Symposium on Lower Power Electronics and Design, pp. 284-289, August 12-14, 2002

Each of these references is included in the Information Disclosure Statement, along with a copy of each reference.

The International Search Report cited the following additional references:
(v) International Publication No. WO 02/103631 A1
(vi) T. Zhang et al., "Joint Code and Decoder Design for Implementation-Oriented (3,k)-Regular LDPC Codes", Proceedings, 35th Asilomar Conference on Signals, Systems, & Computers, pp. 1232-1236, November 4-7, 2001
(vii) E. Boutillon et al., "Decoder-First Code Design", Proceedings, International Symposium on Turbo Codes and Related Topics, pp. 459-462, September 4-7, 2000
(viii) G. Al-Rawi et al., "Optimizing the Mapping of Low-Density Parity Check Codes on Parallel Decoding Architectures", Proceedings, IEEE Conference on Information Technology: Coding and Computing, pp. 578-586, April 2-4, 2001

(ix) A. Selvarathinam et al., “A Massively Scaleable Decoder Architecture for Low-Density Parity-Check Codes”, Proceedings, IEEE International Symposium on Circuits and Systems, Vol. 2, pp. 61-64, May 25-28, 2003

Each of these references is included in the Information Disclosure Statement, along with a copy of each reference. A copy of the International Search Report is also included with the Information Disclosure Statement.

VIII. (E) A discussion of the above-listed references is provided below:

(i) United States Patent Application Publication No. US 2003/0104788 A1 relates to architectures for decoding low density parity check codes that permit varying degrees of hardware sharing to balance throughput, power consumption, and area requirements. The LDPC decoding architectures may be useful in a variety of communication systems in which throughput, power consumption, and area are significant concerns. The decoding architectures implement an approximation of the standard message passing algorithm used for LDPC decoding, thereby reducing computational complexity. Instead of a fully parallel structure, this approximation permits at least a portion of the message passing structure between check and bit nodes to be implemented in a block-serial mode, thus providing reduced area without substantial added latency. The architectures permit varying degrees of hardware sharing to balance throughput, power consumption and area requirements. In addition, in some embodiments, the decoding architectures avoid the need for summations and lookup tables, as typically required by conventional message-passing algorithms. As a result, the decoding architectures can be made more area-efficient.

Specifically, United States Patent Application Publication No. US 2003/0104788 A1 states that due to randomness of connectivity on the bipartite graph representing a parity check matrix, the two classes of computations over a single block of inputs, bit-to-check and check-to-bit, cannot be overlapped. To simplify the control logic, LDPC decoder 44 may incorporate memory is implemented by D flip-flops. The required memory for LDPC decoder 44 in a fully parallel mode, configured to handle (1536,1152) LDPC codes with code rate 3/4, is 5x4992=24960 D flip-flops for storing intermediate messages form check to bit nodes because there exist 4992 edges and message passing on an edge can be represented with a 5-bit value. The memory for LDPC decoder 44 includes a bit-to-check memory and a check-to-bit memory to hold intermediate messages, so the total number of registers required for LDPC is equal to 2x5x4992=49920 D flip-flops with the 5 quantization bits for soft information.

FIG. 20A is a block diagram illustrating a check to bit register 162 in fully parallel mode. FIG. 20B is a block diagram illustrating a check to bit register in half hardware sharing mode, as indicated by registers 164, 166 and demultiplexer 168. FIG. 20C is a block diagram illustrating a check to bit register in 1/k hardware sharing mode, as indicated by registers 170_a through 170_k and demultiplexer 172. Memory from check to bit nodes can be implemented by D flip-flops and demultiplexers as shown in FIGS. 20B and 20C. The memory from check to bit nodes accepts inputs in block-serial and generates outputs in parallel.

In contrast to the present invention, United States Patent Application Publication No. US 2003/0104788 A1 fails to disclose a combination of method and structure for decoding LDPC-coded signals that retrieves edge values associated with a structured parity check matrix used to generate the LDPC-coded signal, wherein the edge values are stored according to a predetermined scheme that permits concurrent retrieval of a set of edge values, and wherein the predetermined scheme preferably specifies contiguous physical memory locations for the set of edge values. There is also no disclosure of edge values having bit nodes of *n* degrees are stored in a first portion of the memory, and edge values having bit nodes of greater than *n* degrees are stored in a second portion of the memory. United States Patent Application Publication No. US 2003/0104788 A1 is also silent on wherein the set of edge values is retrieved in a single clock cycle of a processor coupled to the memory, and is adjacent to a group of M bit nodes or M check nodes, where M is the number of parallel processing engines.

(ii) United States Patent Application Publication No. US 2003/0074626 A1 relates to a method for decoding LDPC codes that comprises executing a sum product algorithm to recover a set of information bits from an LDPC code represented as a bipartite graph of symbol nodes and check nodes. With reference to FIG. 7, a method for decoding Low Density Parity Check (LDPC) codes comprises, at step 200, executing a sum product algorithm to recover a set of information bits from an LDPC code represented as a bipartite graph of symbol nodes and check nodes, the sum product algorithm being responsive to input log likelihood ratios associated with the symbol nodes. The method also comprises, at step 210, updating the check nodes of the sum product algorithm. Referring to FIG. 8, the updating step of the check nodes comprises, at step 220, generating a set of forward difference metrics and a set of backward difference metrics in dependence on the ratios of logarithmic probabilities each associated with a corresponding symbol node of the LDPC code. At step 230, each metric in the set of forward difference metrics is updated in dependence on the absolute value of the log likelihood ratio associated with the symbol node and the absolute value of the previous metric in the set. At step 240, each metric in the set of backward difference metrics is updated in dependence on the absolute value of the log likelihood ratio associated with the symbol node and the absolute value of the previous metric in the set. At step 250, log likelihood ratios to be propagated back to each symbol node are generated in dependence on the updated sets of forward and backward difference metrics.

In contrast to the present invention, United States Patent Application Publication No. US 2003/0074626 A1 fails to disclose a combination of method and structure for decoding LDPC-coded signals that retrieves edge values associated with a structured parity check matrix used to generate the LDPC-coded signal, wherein the edge values are stored according to a predetermined scheme that permits concurrent retrieval of a set of edge values, and wherein the predetermined scheme preferably specifies contiguous physical memory locations for the set of edge values. There is also no disclosure of edge values having bit nodes of *n* degrees are stored in a first portion of the memory, and edge values having bit nodes of greater than *n* degrees are stored in a second portion of the memory. Additionally, United States Patent Application

Publication No. US 2003/0074626 A1 fails to teach wherein the set of edge values is retrieved in a single clock cycle of a processor coupled to the memory, and is adjacent to a group of M bit nodes or M check nodes, where M is the number of parallel processing engines.

(iii) United States Patent Application Publication No. US 2003/0023917 A1 relates to techniques for implementing message passing decoders, for example, LDPC decoders. The authors propose to perform message passing operations serially in time. Incoming messages arrive, e.g., one per clock cycle. It is desirable to have an efficient pipeline structure that can produce one outgoing edge message per clock cycle. To facilitate hardware implementation, messages are quantized to integer multiples of one-half of the natural logarithm of 2. Messages are transformed between more compact variable and less compact constraint node message representation formats. The variable node message format allows variable node message operations to be performed through simple additions and subtractions, while the constraint node representation allows constraint node message processing to be performed through simple additions and subtractions. Variable and constraint nodes are implemented using an accumulator module, subtractor module, and delay pipeline. The accumulator module generates an accumulated message sum. The accumulated message sum for a node is stored and then delayed input messages from the delay pipeline are subtracted therefrom to generate output messages. The delay pipeline includes a variable delay element, thus making it possible to sequentially perform processing operations corresponding to nodes of different degrees.

FIG. 6 depicts a simple serial decoder 600 which performs message processing operations sequentially, one edge at a time. The LDPC decoder 600 comprises a decoder control module 610, a V2C edge memory 630, a C2V edge memory 650, a variable node processor 620, a constraint node processor 640, and output buffer 660. For simplicity, with regard to explaining the invention, we will assume that the decoder runs for a fixed number of iterations, i.e., no convergence detection is performed. V2C and C2V edge memories 630, 650 each include L K bit memory locations with each K bit location corresponding to one edge and where L is the total number of edges in the LDPC graph being used and K is the number of bits per message exchanged along an edge. The output buffer 660 includes memory for storing variable node output values x, which may be either hard (1 bit) or soft (more than 1 bit) values.

In contrast to the present invention, United States Patent Application Publication No. US 2003/0023917 A1 fails to disclose a combination of method and structure for decoding LDPC-coded signals that retrieves edge values associated with a structured parity check matrix used to generate the LDPC-coded signal, wherein the edge values are stored according to a predetermined scheme that permits concurrent retrieval of a set of edge values, and wherein the predetermined scheme preferably specifies contiguous physical memory locations for the set of edge values. There is also no disclosure of edge values having bit nodes of *n* degrees are stored in a first portion of the memory, and edge values having bit nodes of greater than *n* degrees are stored in a second portion of the memory. United States Patent Application Publication No. US 2003/0023917 A1 is also devoid of a teaching of wherein the set of edge values is retrieved in a single clock cycle of a processor coupled to the memory, and is adjacent to a group of M bit nodes or M check nodes, where M is the number of parallel processing engines.

(iv) The article entitled "Low-Power VLSI Decoder Architectures for LDPC Codes" by Mansour et al. relates to an LDPC code and LDPC decoder that are jointly designed to induce the structural regularity needed for a reduced-complexity parallel decoder architecture. The authors note their contributions to be twofold. First, a new interconnect-driven LDPC code design approach is proposed, eliminating the power problem in the interconnection network by inducing desired regularity characteristics in the code. Second, an optimized version of the BCJR algorithm is proposed to compute reliability messages, which minimizes the effect of quantization noise on algorithmic performance and improves the power consumption.

The proposed decoder in Fig. 5 is composed of two main blocks, BLOCK1, the bit-node processing block and BLOCK2, the check-node processing block. Two frames are processed simultaneously by the decoder: phase 2 of the first frame is performed in the check node processing block while phase 1 of the second frame is performed in the bit-node processing block. BLOCK1 contains *c* Bit Function Units (BFU's) and *c* memory banks for storing the check-to-bit messages ($\mu_{c \rightarrow b}$) obtained from BLOCK2 during the previous iteration. Each memory bank consists of *r* memory blocks and a set of *r* counters for address generation in one-to-one correspondence with a single column of *pxp* blocks in the parity check matrix.

In contrast to the present invention, the article entitled "Low-Power VLSI Decoder Architectures for LDPC Codes" fails to disclose a combination of method and structure for decoding LDPC-coded signals that retrieves edge values associated with a structured parity check matrix used to generate the LDPC-coded signal, wherein the edge values are stored according to a predetermined scheme that permits concurrent retrieval of a set of edge values, and wherein the predetermined scheme preferably specifies contiguous physical memory locations for the set of edge values. There is also no disclosure of edge values having bit nodes of *n* degrees are stored in a first portion of the memory, and edge values having bit nodes of greater than *n* degrees are stored in a second portion of the memory. The Mansour et al. article also fails to disclose wherein the set of edge values is retrieved in a single clock cycle of a processor coupled to the memory, and is adjacent to a group of M bit nodes or M check nodes, where M is the number of parallel processing engines.

(v) International Publication No. WO 02/103631 A1 relates to methods and apparatus for decoding code words using message passing decoding techniques which are particularly well suited for use with LDPC codes and long code words. The authors note that there is a need for methods of representing LDPC codes corresponding to large codewords in an efficient and compact manner thereby reducing the amount of information required to represent the code, i.e., to describe the associated graph. The described methods allow decoding graph structures which are largely comprised of multiple identical copies of a much smaller graph. Copies of the smaller graph are subject to a controlled permutation operation to create the larger graph structure. The same controlled permutations are directly implemented to support message passing between the replicated copies of the small graph. Messages corresponding to individual copies of the graph are stored in a memory and accessed in sets, one from each copy of the graph, using a SIMD read or write instruction. The graph permutation operation may be implemented by simply reordering messages, for example, using a cyclic permutation operation, in each set of messages read out of a message memory so that the messages are passed to processing circuits corresponding to different copies of the small graph.

Specifically, the methods allow for decoding of the LDPC graphs that possess a certain hierarchal structure in which a full LDPC graph appears to be, in large part, made up of multiple copies, Z, of a Z times smaller graph. The Z graph copies may be identical. The authors refer to the smaller graph as the projected graph. A decoder implements a sequence of operations corresponding to a message passing algorithm. The same decoder is augmented such that it decodes Z identical such LDPC graphs synchronously and in parallel. Each operation in the message passing algorithm is replicated Z times. The efficiency of the decoding process is improved because, in total, decoding proceeds Z times faster and because the control mechanisms required to control the message passing process need not be replicated for the Z copies but can rather be shared by the Z copies. The process of making Z copies of the smaller graph is viewed as vectorizing the smaller (projected) graph: Each node of the smaller graph becomes a vector node, comprising Z nodes, each edge of the smaller graph becomes a vector edge, consisting of Z edges, each message exchanged in decoding the smaller graph becomes a vector message, comprising Z messages.

In contrast to the present invention, International Publication No. WO 02/103631 A1 fails to disclose a combination of method and structure for decoding LDPC-coded signals that retrieves edge values associated with a structured parity check matrix used to generate the LDPC-coded signal, wherein the edge values are stored according to a predetermined scheme that permits concurrent retrieval of a set of edge values, and wherein the predetermined scheme preferably specifies contiguous physical memory locations for the set of edge values. There is also no disclosure of edge values having bit nodes of *n* degrees are stored in a first portion of the memory, and edge values having bit nodes of greater than *n* degrees are stored in a second portion of the memory. Additionally, the International Publication No. WO 02/103631 fails to disclose wherein the set of edge values is retrieved in a single clock cycle of a processor coupled to the memory, and is adjacent to a group of M bit nodes or M check nodes, where M is the number of parallel processing engines.

(vi) The article entitled "Joint Code and Decoder Design for Implementation-Oriented (3,k)-regular LDPC Codes" by Zhang et al. relates to a proposed joint code and decoder design approach to construct a class of (3,k)-regular LDPC codes which exactly fit to a partly parallel decoder implementation. Each code in this code ensemble is constructed by letting the decoder insert certain random check nodes into the deterministic high-girth (2, *k*)-regular LDPC code. The (3, *k*)-regular decoder (shown in Figure 4) contains k^2 memory banks, the i^{th} memory bank is represented as MEM BANK-(*x*,*y*), where *x* = ((i-1)mod *k*) + 1 and $y = \left[\frac{i-1}{k} \right] + 1$, and each one stores all the intrinsic information (in RAM *I*), extrinsic information (in two-port RAM *E*1, *E*2, and *E*3) and estimated decoded bits (in RAM *C*) associated with *L* variable nodes; a 1-layer shuffle network (π_{-1} or Id); a g-layer shuffle network; *k* Check Node processor Units (CNU's) and k^2 Variable Node processor Units (VNU's). One Address Generator (AG) is associated with each memory bank to provide the access address.

The authors have observed that the straightforward fully parallel decoder architecture usually incurs too high a complexity for many practical purposes, and therefore should be transformed to a partly parallel realization. The partly parallel decoder architecture presented in the article is suitable for VLSI implementation, and it has been shown that the jointly developed (3,k)-regular LDPC codes have very good performance.

In contrast to the present invention, the Zhang et al. article entitled "Joint Code and Decoder Design for Implementation-Oriented (3,k)-regular LDPC Codes" fails to disclose a combination of method and structure for decoding LDPC-coded signals that retrieves edge values associated with a structured parity check matrix used to generate the LDPC-coded signal, wherein the edge values are stored according to a predetermined scheme that permits concurrent retrieval of a set of edge values, and wherein the predetermined scheme preferably specifies contiguous physical memory locations for the set of edge values. There is also no disclosure of edge values having bit nodes of *n* degrees are stored in a first portion of the memory, and edge values having bit nodes of greater than *n* degrees are stored in a second portion of the memory. The Zhang et al. also fails to disclose wherein the set of edge values is retrieved in a single clock cycle of a processor coupled to the memory, and is adjacent to a group of M bit nodes or M check nodes, where M is the number of parallel processing engines.

(vii) The article entitled "Decoder-First Code Design" by E. Boutillon et al. relates to reversing the sequence order to code design by defining the hardware structure of the decoder and then constructing the code. It is proposed to first choose an efficient hardware structure, and then in the second step, to construct a code that adequately fits the chosen structure. An example of such a methodology is given for an LDPC code. The decoding process, as well as the architecture, is defined prior to the code construction. It is composed of *D* decoding iterations. Each decoding iteration is processed in *M* clock cycles. At every clock cycle, the 5 following operations are performed. First, *N* data is retrieved from the memory banks at the addresses given by the *N* Random Address Generators (RAG). Second, the *N* data is shuffled according to the RPG index. Third, the MAP algorithm is performed on the *P* parity checks. The result is the extrinsic information associated to each data. Fourth, the *N* extrinsic information computed by the parity checks is unshuffled. Fifth, the unshuffled data is stored into the memory bank at their initial location. The code characteristics are derived from the hardware parameters.

In contrast to the present invention, the E. Boutillon et al. article entitled "Decoder-First Code Design" fails to disclose a combination of method and structure for decoding LDPC-coded signals that retrieves edge values associated with a structured parity check matrix used to generate the LDPC-coded signal, wherein the edge values are stored according to a predetermined scheme that permits concurrent retrieval of a set of edge values, and wherein the predetermined scheme preferably specifies contiguous physical memory locations for the set of edge values. There is also no disclosure of edge values having bit nodes of *n* degrees are stored in a first portion of the memory, and edge values having bit nodes of greater than *n* degrees are stored in a second portion of the memory. The E. Boutillon et al. article also fails to teach wherein the set of edge values is retrieved in a single clock cycle of a processor coupled to the memory, and is adjacent to a group of M bit nodes or M check nodes, where M is the number of parallel processing engines.

(viii) The article entitled "Optimizing the Mapping of Low-Density Parity Check Codes on Parallel Decoding Architectures" by G. Al-Rawi et al. relates to a study of the problem of optimizing the mapping of LDPC codes on parallel machines to minimize the communication cost. The authors attempt to address the problem of determining the best mapping of logical variable and function nodes to physical nodes that will minimize overall execution time and communication cost. To reduce the search space, the problem is solved in two stages: clustering, and cluster allocation. A simplified clustering technique based on a modified min-cut algorithm that reduces the search complexity from $O(n^2)$ to $O(n)$ is proposed. The proposed solution is based on mapping almost equal number of logical function and variable nodes to each physical node. By imposing restrictions on the configuration space, the mapping problem reduces to allocating appropriate logical nodes of a predetermined type and number to each physical node so as to reduce the total communication cost.

In contrast to the present invention, the article entitled "Optimizing the Mapping of Low-Density Parity Check Codes on Parallel Decoding Architectures" fails to disclose a combination of method and structure for decoding LDPC-coded signals that retrieves edge values associated with a structured parity check matrix used to generate the LDPC-coded signal, wherein the edge values are stored according to a predetermined scheme that permits concurrent retrieval of a set of edge values, and wherein the predetermined scheme preferably specifies contiguous physical memory locations for the set of edge values. There is also no disclosure of edge values having bit nodes of *n* degrees are stored in a first portion of the memory, and edge values having bit nodes of greater than *n* degrees are stored in a second portion of the memory. In addition, the Al-Rawi et al. article provides no mention of wherein the set of edge values is retrieved in a single clock cycle of a processor coupled to the memory, and is adjacent to a group of M bit nodes or M check nodes, where M is the number of parallel processing engines.

(ix) The article entitled "A Massively Scaleable Decoder Architecture for Low-Density Parity-Check Codes" by A. Selvarathinam et al. relates to a massively scaleable architecture for decoding LDPC codes. The authors propose a pseudo-random construction of the parity check matrix that is specifically designed to facilitate parallelization of the decoder. Parallelization is made possible by simultaneously processing M bit nodes and making certain that each of the M consecutive bit nodes are associated with different parity checks. Each group of parity check equations is a separate block of memory that can independently read and write in every clock cycle. 3*M memory blocks are required to process all the three edges of M consecutive bit nodes. Thus memory is partitioned by a factor of M.

In contrast to the present invention, the article entitled "A Massively Scaleable Decoder Architecture for Low-Density Parity-Check Codes" fails to disclose a combination of method and structure for decoding LDPC-coded signals that retrieves edge values associated with a structured parity check matrix used to generate the LDPC-coded signal, wherein the edge values are stored according to a predetermined scheme that permits concurrent retrieval of a set of edge values, and wherein the predetermined scheme preferably specifies contiguous physical memory locations for the set of edge values. There is also no disclosure of edge values having bit nodes of *n* degrees are stored in a first portion of the memory, and edge values having bit nodes of greater than *n* degrees are stored in a second portion of the memory. The Selvarathinam et al. article also fails to disclose wherein the set of edge values is retrieved in a single clock cycle of a processor coupled to the memory, and is adjacent to a group of M bit nodes or M check nodes, where M is the number of parallel processing engines.

CONCLUSION

It is respectfully requested that examination of the above-referenced application be advanced in accordance with the provisions of 37 C.F.R. § 1.102 and MPEP 708.02.

Applicants' undersigned attorney may be reached by telephone at (301) 601-7252. All correspondence should continue to be directed to our address given below.

Respectfully submitted,

August 6, 2004

Craig L. Plastrik
Registration No. 41,254

Hughes Electronics Corporation
Customer No. 20991